NAME (as it appears on your UF ID): ____________________________

(Please **PRINT**)

UF Student ID#: _______________

--------------------- CEN 4072/6070 Software Testing & Verification ---------------------

Exam 2 – Spring 2015

You have 90 minutes to work on this exam. It is a "closed-book/closed-notes" test.
Pay attention to point values, since you may not have time to complete all 12 problems.
You should assume that all variables represent INTEGERS, unless otherwise indicated.

PRINT your name above NOW and sign the pledge at the bottom of the last page, if appropriate, when you are finished.

<u>PLEASE PRINT ANSWERS IN THE SPACE PROVIDED</u> **<u>(EXCLUDING MARGINS)</u>** <u>ONLY – PREFERABLY USING A BALLPOINT PEN TO INCREASE LEGIBILITY</u>. Good luck!

1. (16 pts.) Consider the assertion of ***weak*** correctness: {t=5 Λ z<0} S {y=z+1 Λ t=z}. Which of the following observations/facts **would** allow one to deduce that the assertion is false, and which **would not**? Circle either "would" or "would not" as appropriate, considering the observations individually. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

   a. When the product of t and z is equal to -5 prior to the execution of S, t is NOT equal to z when s terminates. would would not

   b. When the initial value of t is 5 and the initial value of z is -1, the value of t is 17 when S terminates. would would not

   c. When the initial value of t is 5 and the initial value of z is -17, S does not terminate. would would not

   d. Whenever the initial value of z is -7, the value of y is sometimes less than the value of z when S terminates. would would not

   e. wp(S, t=z) = |t|>2 would would not

   f. wp(S, y=z) = z>-5 would would not

   g. wlp(S, y=z) = z>-5 would would not

   h. sp(S, t=5) = t=z would would not

2. (20 pts.) Circle either “true” or “false” for each of the following assertions. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. $[\{x>0\}\ S\ \{x>0\}] \Rightarrow [\{x>17\}\ S\ \{x=5\}]$ true false

b. {x=10} while x<>5 do x := x-3 {x=5} true false

c. $[sp(S, x\geq 5) = (y=17)] \Rightarrow [\{x=5 \wedge z=0\}\ S\ \{y>0\}]$ true false

d. Formally speaking, Z=XJ is a **loop invariant** for the assertion: true false

```
{Z=0 Λ J=0}
     while J<>Y do
          Z := Z+X;
          J := J+1
     end_while
{Z=XY}
```

e. $Z=XJ \wedge J\geq 0$ is a **Q-adequate loop invariant** for the assertion given in part (d) above. true false

f. The following program terminates for ***all*** initial integer values of x,y. true false

```
while (y>0) do
   y := y+x
   if (x≥0) then
        x := x-1
   end_if
end_while
```

g. The Method of Well Founded Sets, as presented in class, can be used to prove that the program given in part (f) above will terminate. true false

h. {P} while b do S {Q} ⇔ true false

```
{P}
    if b then
        S
    end_if
    while b do
        repeat S until NOT b
    end_while
{Q}
```

i. Suppose k = wp(while b do S, Q). Then k is a Q-adequate loop invariant for {P} while b do S {Q} for any P. true false

j. Suppose sp(S,k) = t. Then $k \Rightarrow wlp(S, t)$. true false

3. a. (2 pts.) Complete the ROI for proving the strong correctness of program S with respect to pre-condition P and post-condition Q using the weakest pre-condition (wp) predicate transform:

$$\frac{\rule{6cm}{0pt}}{\{P\}\ S\ \{Q\}\ \text{strongly}}$$

b. (14 pts.) Find the weakest pre-condition (wp) of the program:

while x<>0 do x := x-1; y := y+3 end-while

with respect to the post-condition {y=17}. (Give the values of H_0, H_1, H_2, and H_k, where the wp is given by the infinite expression: $H_0 \vee H_1 \vee H_2 \vee \ldots \vee H_k \vee \ldots$ and then express the wp in closed form in terms of x and y only.)

H_0:

H_1:

H_2:

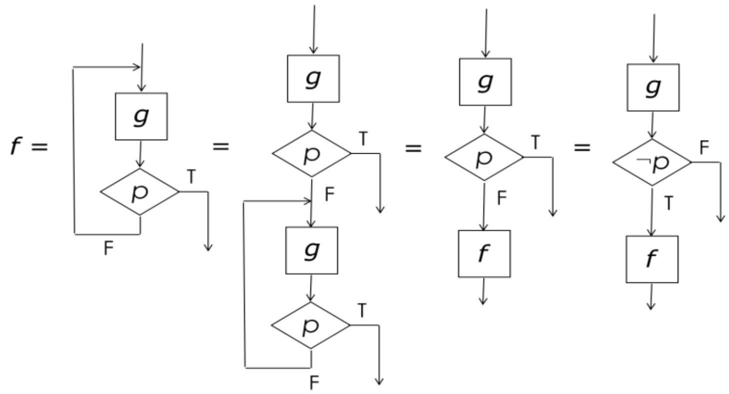
H_k:

Closed form expression of wp: ____________________

c. (2 pts.) Use the ROI from part (a) and the closed form expression of the wp from part (b) to prove the assertion:

{ $x=5 \wedge y=2$ } while x<>0 do x := x-1; y := y+3 {y=17} strongly

4. (3 pts.) The diagram below was used in class to illustrate an important concept/result related to functional verification.



Which one of the following concepts/results was derived in connection with the control flow relationships illustrated? (Circle ONE only.)

a. the Rule of Inference for proving {P} *repeat g until p* {Q}

b. the functional relationship between repeat_until and while_do constructs

c. the weakest possible *f*-adequate loop invariant for [*repeat g until p*]

d. the Axiom of Replacement

e. the weakest pre-condition of *repeat g until p* with respect to post-condition Q

f. the complete correctness conditions for *f* = [*while ¬p do g*]

g. (none of the above)

5. (8 pts.) Given *P1, P2, f1,* and *f2:*

P1 = repeat x := x-1; z := z∗x until x<=2

P2 = repeat x := x-1; z := z∗x until x=1

f1 = (x=2 -> x, z := 1, z | x≤1 -> x,z := x-1,zx-z)

f2 = (x>1 -> x, z := 1, z(x-1)!)

Determine the correctness relationships among the given programs and functions. In the table below, indicate "C" for Complete program correctness, "S" for Sufficient program correctness only, and "N" for Neither. To compensate for random guessing, you will receive +2 pts. for each correct answer given and -1 pt. for each incorrect answer given, with a minimum possible score of 0 pts.

	P1	*P2*
f1		
f2		

6. (20 pts.) Circle either "valid" or "invalid" for each of the following *hypothesized* Rules of Inference. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. $\dfrac{\{P\}\ \text{if } b \text{ then } S\ \{(b \vee Q) \wedge P\}}{\{P\}\ \text{while } b \text{ do } S\ \{Q\}}$? valid invalid

b. $\dfrac{sp(S, P) \Rightarrow \text{true}}{\{P\}\ S\ \{Q\}}$? valid invalid

c. $\dfrac{(\neg b) \Leftrightarrow Q}{\{P\}\ \text{while } b \text{ do } S\ \{Q\} \text{ strongly}}$? valid invalid

d. $\dfrac{Q \Leftrightarrow sp(S, \text{true})}{\{P\}\ S\ \{Q\}}$? valid invalid

e. $\dfrac{\{P \wedge b\}\ S\ \{I\},\ ((I \vee P) \wedge \neg b) \Rightarrow Q}{\{P\}\ \text{while } b \text{ do } S\ \{Q\}}$? valid invalid

f. $\dfrac{(wlp(S, Q) \wedge K) \Leftrightarrow P}{\{P\}\ S\ \{Q\}}$? valid invalid

g. $\dfrac{\{\text{true}\}\ S\ \{I\},\ (I \wedge b) \Rightarrow Q}{\{P\}\ \text{repeat } S \text{ until } b\ \{Q\}}$? valid invalid

h. $\dfrac{P \Leftrightarrow I,\ \{I\}\ S\ \{I\},\ P \Rightarrow Q}{\{P\}\ \text{repeat } S \text{ until } b\ \{Q\}}$? valid invalid

i. $\dfrac{\{\text{true}\}\ S\ \{K\},\ \{K\}\ \text{while } \neg b \text{ do } S\ \{Q\}}{\{P\}\ \text{repeat } S \text{ until } b\ \{Q\}}$? valid invalid

j. $\dfrac{\{\text{true}\}\ \text{if } b \text{ then } S\ \{Q\} \text{ strongly}}{\{P\}\ \text{while } b \text{ do } S\ \{Q\} \text{ strongly}}$? valid invalid

7. (11 pts.) Prove the assertion of weak correctness below using the Repeat_Until Rule of Inference with the invariant: $t=2^{k}$. SHOW AND JUSTIFY ALL STEPS AND CASES AS ILLUSTRATED IN CLASS.

$\{n \geq -17\}$

```
t := 1
k := 0
repeat
    t := 2*t
    k := k+1
until k=n
```

$\{t=2^{n}\}$

8. (21 pts.) For program *H* and intended program function *f* given below, Prove *f* = [*H*] by showing that the repeat_until complete correctness conditions hold. You may assume (i.e., you need NOT prove) that the function of the loop body, *g*, is (t,k := 2t,k+1). STATE AND **PROVE** ALL OTHER CONDITIONS, STEPS, AND CASES AS ILLUSTRATED IN CLASS.

H:

```
repeat
   t := 2*t
   k := k+1
until k=n
```

$f = (k<n \rightarrow t,k := t2^{n-k},n)$

(Continue your proof on the next page if necessary.)

8. (cont’d)

9. a. (10 pts.) Assume that for “input” $X_0 = (y_0, z_0)$, a while loop computing function *t* terminates after *n* iterations with “output” $X_n = (y_n, z_n)$. Furthermore, assume X_1, X_2, ..., X_k, ..., X_{n-1} are the intermediate states generated by the loop, and that $t(X_k) = (0, z_0-3y_0)$.

   i. What are the values of y_n and z_n (in terms of z_0 and y_0)?

   ii. What is $t(X_n)$?

   iii. Give the weakest *t*-adequate invariant for *any* while loop computing this function.

b. (4 pts.) Suppose it was determined that $t(X_k) = (0, -1)$. Could one then deduce a unique X_0 from this fact? **If so**, what would the unique values of y_0 and z_0 be? **If not**, what *could* be deduced about y_0 and z_0? (Hint: could X_0 be (6,17)? (-3,10)? (12,35)?)

c. (4 pts.) Is there a program of the form *while b do S* that could compute *t* and produce intermediate state (5,-7) for some input? (Circle ONE only.)

   i. Yes, every program of the form *while b do S* that computes *t* would produce intermediate state (5,-7) for some input.

   ii. No, since for any input, the intermediate state (5,-7) is inconsistent with the invariant *q*(X) for *t*.

   iii. Yes, there could be a program of the form *while b do S* that computes *t* and produces intermediate state (5,-7) for some input *provided* that the program terminates with y=0 and z=-22.

   iv. Yes, every program of the form *while b do S* that computes *t* and terminates with y=0 and z=-22 would produce intermediate state (5,-7) for some input.

   v. (None of the above – this cannot be determined based on the info provided.)

d. (4 pts.) Is there a program of the form *while b do S* that computes *t* and produces both intermediate states (5,-7) and (3,8) for a given input? Briefly but carefully justify your answer.

10. (10 pts.) The following statements relate to King, et al., "Is Proof More Cost Effective than Testing?" Indicate whether each is true or false. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. The application described in the paper, "SPARK", is a Z-specified compiler implemented in a subset of Ada that is used for procuring safety critical software for defense systems.	true	false
b. When the customers reviewed a sample of the Z proofs (selected by them), only typographical errors were found.	true	false
c. A "lesson learned" by the authors was that at the "top" level of the system, proof annotations were often too large to be manageable, while at the "bottom" there was a need to interface with software (such as device drivers) for which there was *no* formal specification.	true	false
d. Code proofs made use of predicate transforms for non-looping programs, while invariants plus separate arguments for termination were used for loops.	true	false
e. Substantial amounts of unit testing were completed before the bulk of code proof started.	true	false

11. (10 pts.) The following statements relate to Linger, "Cleanroom Software Engineering for Zero-Defect Software." Indicate whether each is true or false. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. The Cleanroom process is based on the philosophy that *developers need to be comfortable with testing less. "The key aspect is to balance...the feature set and (their) quality... If it is too high, then you won't be competitive in the market place and your code won't be exercised."*	true	false
b. Cleanroom management planning and control is based on developing and certifying a pipeline of software increments that accumulate to the final product.	true	false
c. A significant advantage of testing based on the expected frequency of user executions (from an operational profile) over coverage testing is that flaws in high-consequence functions will be found first.	true	false
d. The system-level test team is responsible for "measuring quality" using error seeding and mutation analysis.	true	false
e. Linger notes that while the Cleanroom process is readily applied to new systems development, re-engineering and extension of existing systems require the use of other, more traditional processes.	true	false

12. (12 pts.) Use the Axiom of Replacement and function composition to *deduce* the function of the following program:

```
y := x-1
if y>0 then
   x := x+1
else
   y := -y
end_if_else
x := 3
```

Express the function as: ($p_1 \rightarrow$ x,y := ?,? | $p_2 \rightarrow$ x,y := ?,?) where p_1 and p_2 are Boolean predicates, the union of which specifies the function domain.

On my honor, I have neither given nor received unauthorized aid on this exam and I pledge not to divulge information regarding its contents to those who have not yet taken it.

____

SIGNATURE